INTERNAL ASSESSMENT TEST – II Solution
Sub: Computer Organization and ARM Microcontrollers (21EC52)

USN										

INTERNAL ASSESSMENT TEST – II

Sub:	Computer Organization and ARM Microcontrollers							Code:	21EC52
Date:	30/01/24	Duration:	90 mins	Max Marks:	50	Sem:	V	Branch:	ECE

Answer any 5 full questions

		Marks	CO	RBT
1	(a) Compare CISC and RISC processors. (b) Give notes on AMBA Bus protocol.	[5] [5]	CO3	L2
2	With the help of neat block diagram, explain Embedded system hardware in detail.	[10]	CO3	L2
3	Explain the different processor modes of the ARM processor with the help of relevant diagrams.	[10]	CO3	L2
4	With the help of neat block diagram, Explain the data flow of the ARM processor.	[10]	CO3	L2

		Marks	CO	RBT
5	Explain the pipeline concept in ARM Processor and its execution sequence with the help of neat diagram.	[10]	CO1	L2
6	Explain in detail about three different hardware core extensions for ARM processors.	[10]	CO3	L2
7	Give the control sequence for the execution of the following instructions (a) MOVS r0, r1, LSL #1 (b) RSB r0, r1, #0	[5] [5]	CO3	L3
8	Give the control sequence for the execution of the following instructions (a) UMULL r0, r1,r2,r3 (b) LDMIA r0!, {r1-r3}	[5] [5]	CO3	L3

CI CCI

HOD

(a) Compare CISC and RISC processors.
(b) Give notes on AMBA Bus protocol.

Differentiate CISC and RISC

RISC	CISC
It is a Reduced Instruction Set Computer.	It is a Complex Instruction Set Computer.
It focuses on Software.	It focuses on Hardware.
It uses only a Hardwired control unit.	It uses both hardwired and microprogrammed control units.
Transistors are used for more registers.	Transistors are used for storing complex instructions.
Code size is large.	Code size is small.
The uses of the pipeline are simple in RISC.	Uses of the pipeline are difficult in CISC.
An instruction is executed in a single clock cycle.	Instruction may take more than one clock cycle.
An instruction can fit in one word.	Instructions are larger than the size of one word.
The execution time of RISC is very short.	The execution time of CISC is longer.
The program written for RISC architecture needs to take more space in memory.	The Program written for CISC architecture tends to take less space in memory.

1.3.2 AMBA

- AMBA Advanced Micro controller Bus Architecture
  - 1996, it's introduced and widely adopted as the on-chip bus architecture for ARM processors.
  - The first AMBA buses introduced were
    - » ASB : ARM System Bus, and
    - » APB : ARM Peripheral Bus
  - Later, ARM introduced another bus design
    - » AHB: ARM High-performance Bus

- Using AMBA,
  - peripheral designers can reuse the same design on multiple projects (with different processor architecture).
  - Plug-and-play

AHB

- AHB
  - provides higher data throughput than ASB. Because
    - » It use a Centralized Multiplexed Bus Scheme (rather than ASB's bidirection bus).
    - » This change allows the AHB bus to run at higher clock speed.
    - » 64/128 bits width.

- Two variations on the AHB bus
  - » Multi-layer AHB, and
    - allows multiple active bus masters,
  - » AHB-Lite: only one master

2. With the help of neat block diagram, explain Embedded system hardware in detail.

Embedded System Hardware

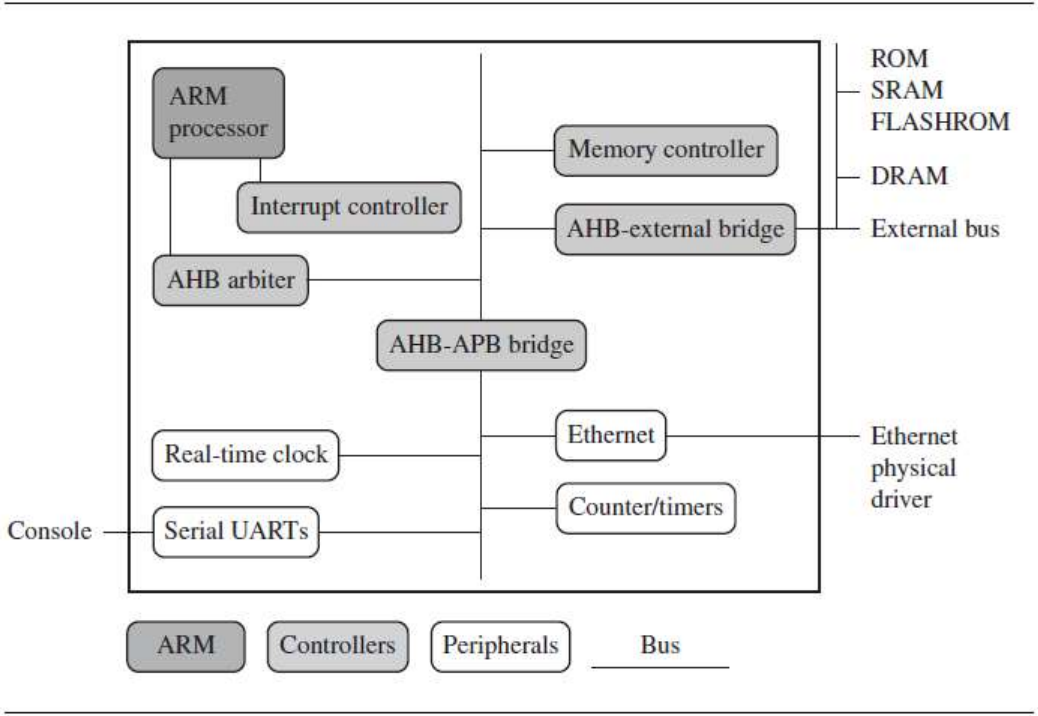


Figure 2: An Example of an ARM-based embedded device, a microcontroller

Units inside SoC

- SoC is an embedded device.
- We can separate the device into four main components:
  - ARM Processor: controls the embedded device.
    - » An ARM processor comprises a core (the execution engine that processes instructions and manipulates data), plus the surrounding components (MMU and caches) that interface it with a bus.
  - Controllers: coordinate important functional blocks (e.g. interrupt and memory controllers)
  - Peripherals: USB, LCD, etc.
  - Bus: is used to communicate between different parts of the device.

1.3.1 ARM Bus Technology

- Embedded systems use different bus technologies than those designed for x86 PC.
  - Embedded device use an on-chip bus
  - Core is master who initiates a data transfer.
- A Bus has two architecture levels
  - The First is a physical level that covers the *electrical characteristics* and *bus width* (16, 32, or 64 bits).
  - The Second level deals with *protocol*.- the logical rules governing the communication between processor and peripheral.
- ARM seldom implements the electrical characteristics of the bus, but it routinely specifies the bus protocol.

1.3.2 AMBA

- AMBA Advanced Micro controller Bus Architecture
  - 1996, it's introduced and widely adopted as the on-chip bus architecture for ARM processors.
  - The first AMBA buses introduced were
    - » ASB : ARM System Bus, and
    - » APB : ARM Peripheral Bus
  - Later, ARM introduced another bus design
    - » AHB: ARM High-performance Bus

- Using AMBA,
  - peripheral designers can reuse the same design on multiple projects (with different processor architecture).
  - Plug-and-play

AHB

- AHB
  - provides higher data throughput than ASB. Because
    - » It use a Centralized Multiplexed Bus Scheme (rather than ASB's bidirection bus).
    - » This change allows the AHB bus to run at higher clock speed.
    - » 64/128 bits width.

- Two variations on the AHB bus
  - » Multi-layer AHB, and
    - allows multiple active bus masters,
  - » AHB-Lite: only one master

1.3.3 Memory

- **Memory is necessary**
  - An embedded system has to have some form of memory to store and execute code.
- **You have to consider**
  - price, performance, and power consumption
- **Specific memory characteristics**
  - hierarchy, width, and type

Memory Hierarchy

- Cache
  - is used to speed up data transfer between Core and Main Memory (DRAM);
- But,
  - It makes the performance unpredicted;
  - It doesn't help Real-Time system response;
    - » Note that many small embedded systems do not require the benefit of a cache.
- * Cache
  - Elastic buffer (different speed between Core and Bus);
  - Width adaptive (e.g., 32-bit Core vs. 16-bit BUS)

Memory Types

- **DRAM**
  - the most commonly used RAM for devices;
  - Dynamic: need to have its storage cells refreshed and given a new electronic charge every few milliseconds, so you need to set up a DRAM controller before using the memory.
- **SRAM**
  - is faster than the more traditional DRAM (SRAM does not require a pause between data access).
- **SDRAM**
  - is one of many subcategories of DRAM.
  - accessed pipelined, transferred in a burst.

1.3.4 Peripherals

- Embedded system that interact with the outside world need some form of peripheral device.
  - Peripherals range from a simple serial communication device to a more complex 802.11 wireless device.

- All ARM peripherals are memory mapped – the programming interface is a set of memory addressed register.

- Controllers are specialized peripherals that implement higher level of functionality within an embedded system.
  - Two important types of controllers are
    - » Memory Controller
    - » Interrupt Controller
      - Normal IC
      - Vectoring IC
        - Priority
        - Simple Interrupt Dispatch

Memory Controllers

- **Memory Controllers: Connect different types of memory to the processor bus.**
  - On power-up a memory controller is configured in hardware to allow certain memory device to be active. These memory devices allow the initialization code to be executed.
  - Some memory devices must be set up by software.
    - » e.g. When using DRAM, you first have to set up the memory timings and refresh rate before it can be accessed.

Interrupt Controller

- When a peripheral or device requires attention,
  - it raise an interrupt to the processor.
- An interrupt controller
  - provides a programmable governing policy

- There are two types of interrupt controller available for the ARM processor

  - Standard interrupt controller
    - » Sends an interrupt signal; Can be programmed to ignore or mask an individual or set of devices.
    - » It's interrupt handler determines which device requiring service.

  - Vector interrupt controller (VIC)
    - » Associate a "priority" and a "handler address" to each interrupt.
    - » Depending on its type, VIC will either call the standard interrupt exception handler (loading the handler address from VIC) or cause core to jump to the handler for the device directly.

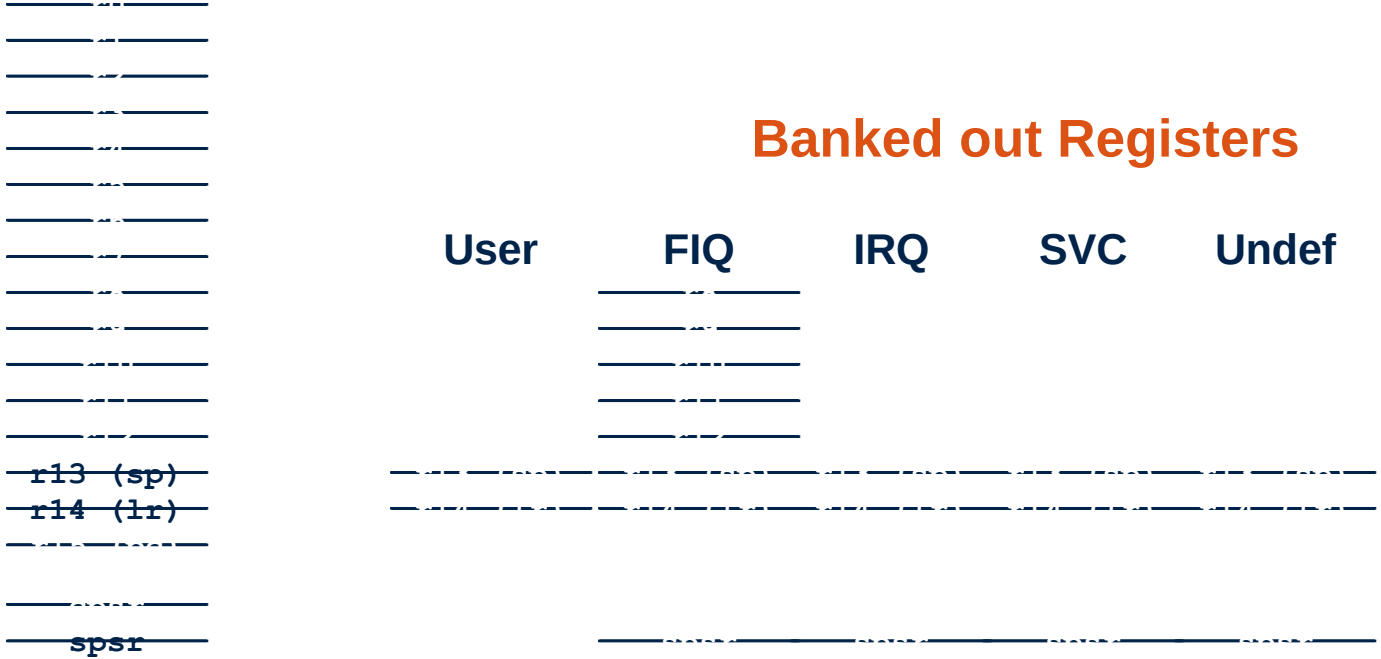
3. Explain the different processor modes of the ARM processor with the help of relevant diagrams.

Processor Modes

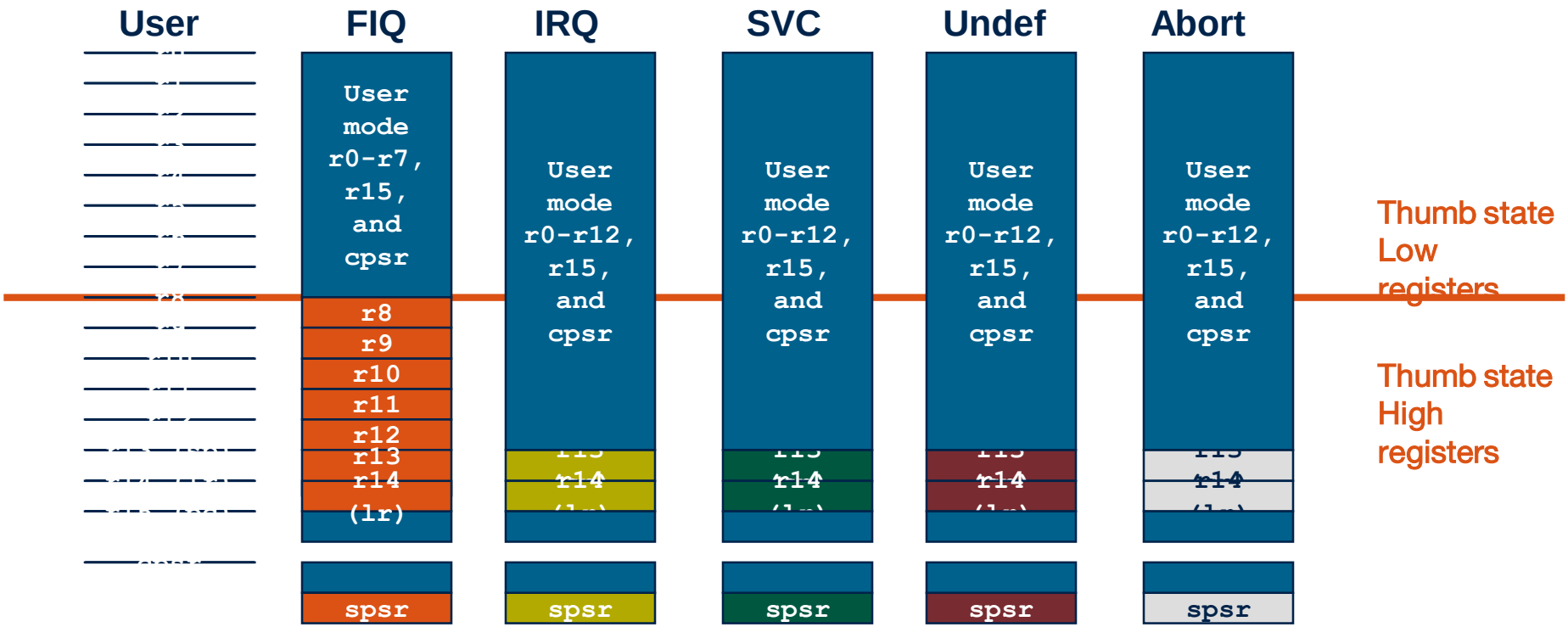
- **User** : unprivileged mode under which most tasks run
- **FIQ** : entered when a high priority (fast) interrupt is raised
- **IRQ** : entered when a low priority (normal) interrupt is raised
- **Supervisor** : entered on reset and when a Software Interrupt instruction is executed
- **Abort** : used to handle memory access violations
- **Undef** : used to handle undefined instructions
- **System** : privileged mode using the same registers as user mode

The ARM Register Set

Current Visible Registers



Register Organization Summary



Note: System mode uses the User mode register set

4. With the help of neat block diagram, Explain the data flow of the ARM processor.

Data-path in ARM

- ALU
  - Sources: Rn, Rm(shifted);
  - Destination: Rd
- Shifter
  - Helps to Extend the scope of data or address
- Sign-extend
  - LOAD data from Main memory

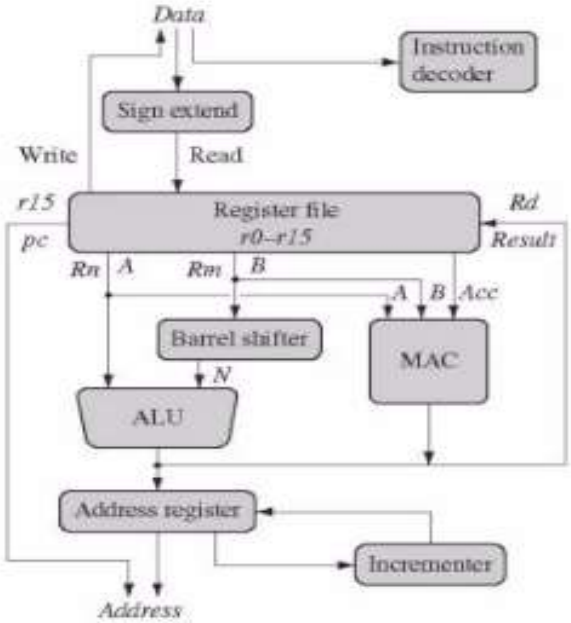


Figure 2.1 ARM core dataflow model.

5. Explain the pipeline concept in ARM Processor and its execution sequence with the help of neat diagram.

PipeLine

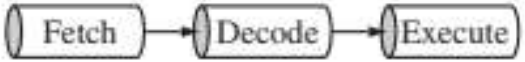


Figure ARM7 Three-stage pipeline.

PipeLine

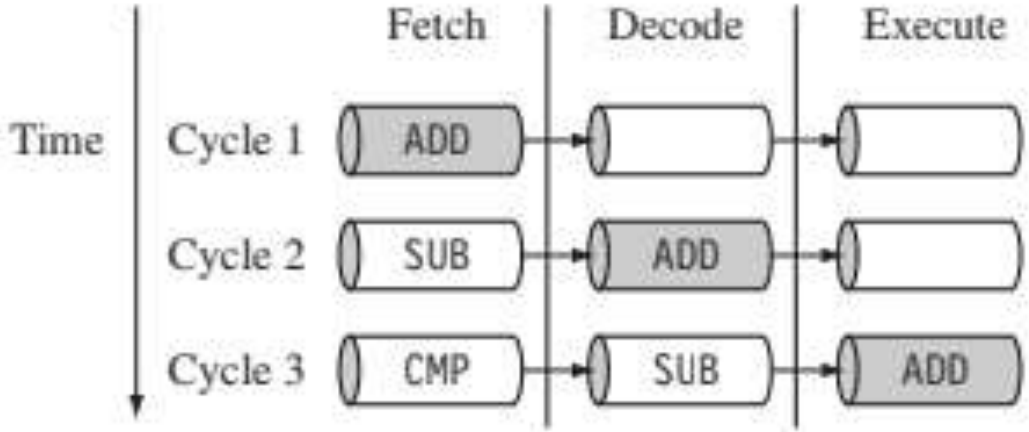


Figure Pipelined instruction sequence.

PipeLine

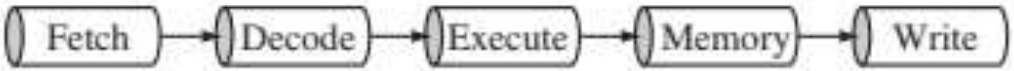


Figure ARM9 five-stage pipeline.

Figure ARM10 six-stage pipeline.

Pipeline Executing Characteristics

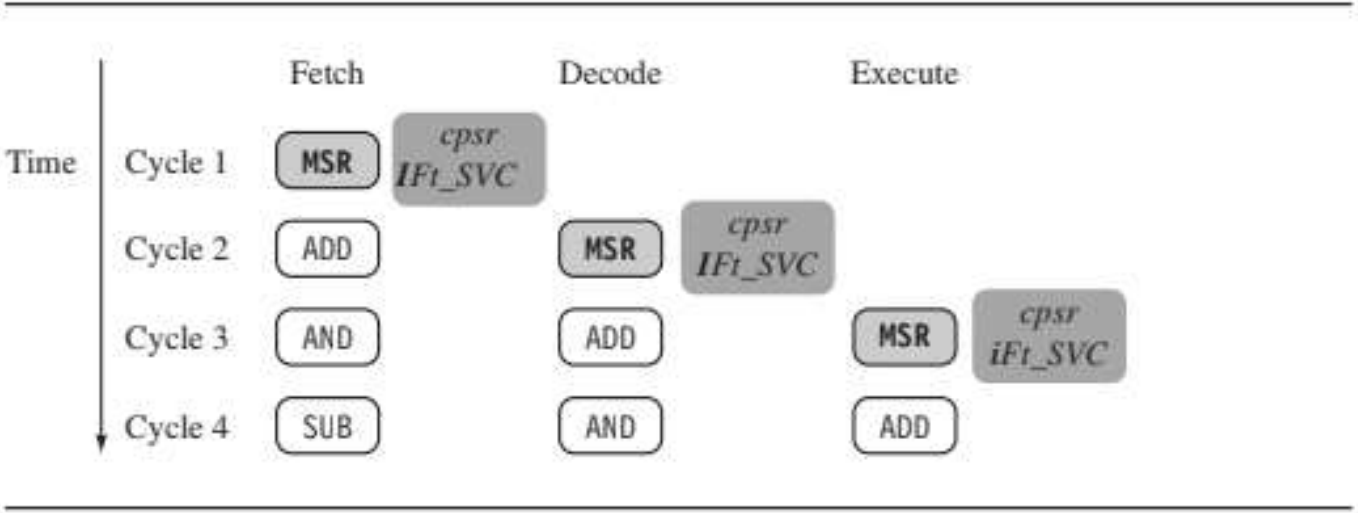


Figure 2.11 ARM instruction sequence.

Pipeline Executing Characteristics

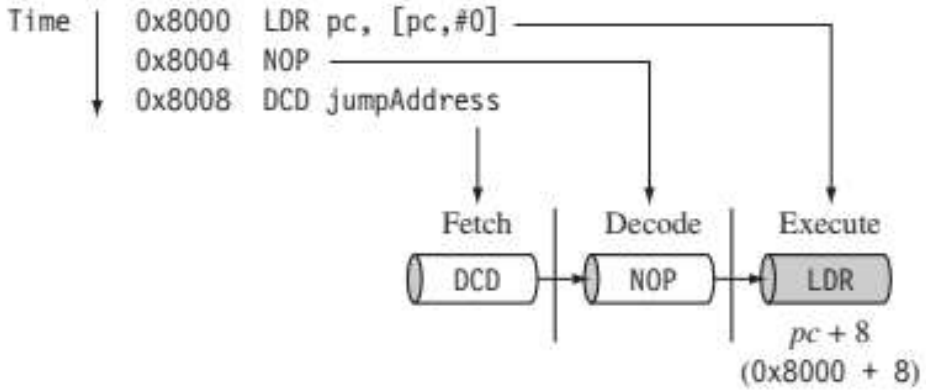


Figure 2.12 Example: pc = address + 8.

6. Explain in detail about three different hardware core extensions for ARM processors.

Core Extensions

(i) Cache and Tightly Coupled Memory

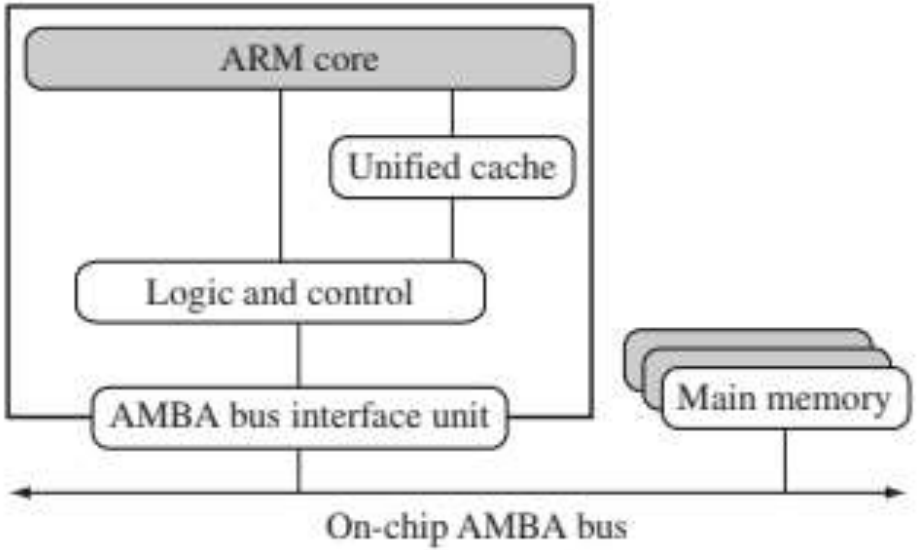


Figure A simplified Von Neumann architecture with cache.

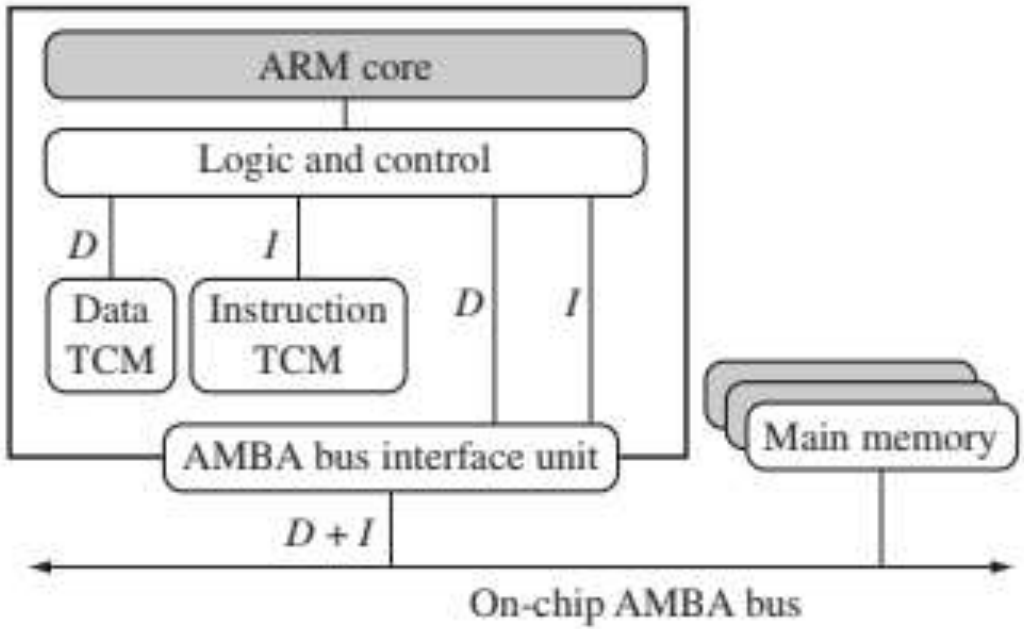


Figure A simplified Harvard architecture with TCMs.

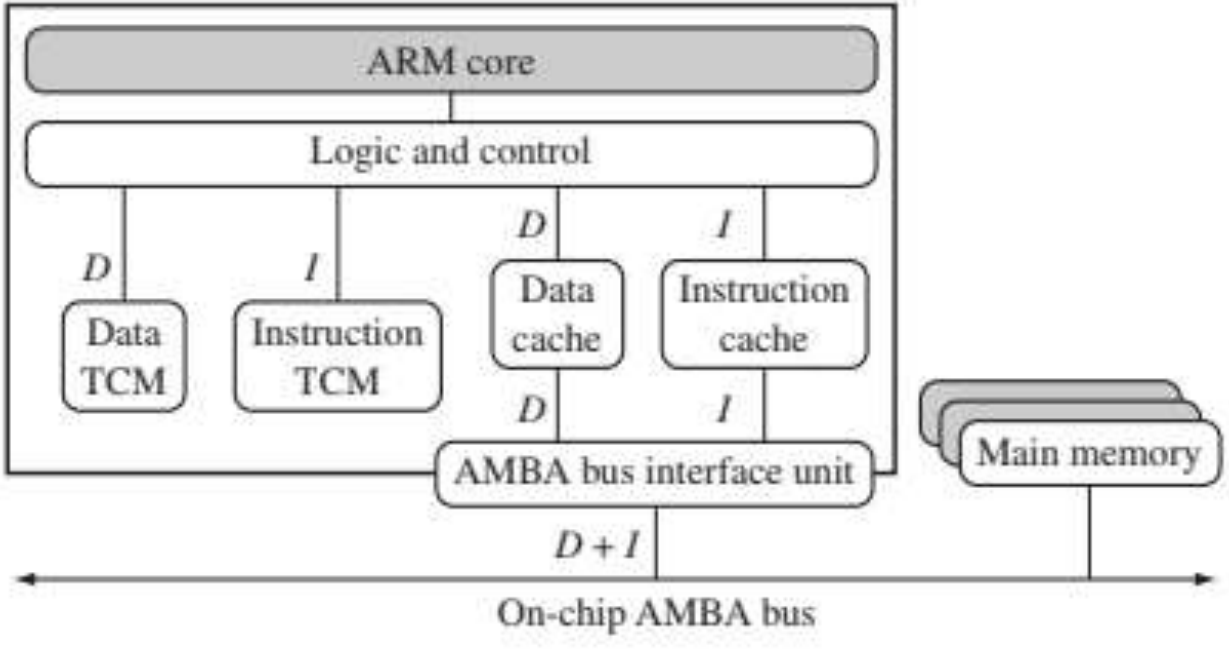


Figure: A simplified Harvard architecture with caches and TCMs.

(ii) Memory Management

- Non Protected Memory
- Memory Protection Unit
- Memory Management Unit

(iii) Coprocessors

- Coprocessors can be attached to the ARM processor. A coprocessor extends the processing features of a core by extending the instruction set or by providing configuration registers.
- More than one coprocessor can be added to the ARM core via the coprocessor interface.
- The coprocessor can be accessed through a group of dedicated ARM instructions that provide a load-store type interface.
- Consider, for example, coprocessor 15: The ARM processor uses coprocessor 15 registers to control the

7. Give the control sequence for the execution of the following instructions

(a) MOVS r0, r1, LSL #1

(b) RSB r0, r1, #0

- ***Example 2***
  - PRE: r0 = 0x00000000, r1 = 0x80000004
  - MOVs r0, r1, LSL #1 ; r0 = r1 *2
  - POST r0 = **0x00000008**, r1 = 0x80000004

- RSB : reverse subtract
  - RSB r0, r1, r2; r0 = r2 – r1

8. Give the control sequence for the execution of the following instructions

(a) UMULL r0, r1,r2,r3

(b) LDMIA r0!, {r1-r3}

- UMULL : unsigned multiply long
  - UMULL r0, r1, r2, r3; [r1,r0] = r2*r3

Multiple-Register Transfer (Cont.)

- ***Example 9***
  - **PRE:**

    mem32[0x80018] = 0x03,

    mem32[0x80014] = 0x02,

    mem32[0x80010] = 0x01, r0 = 0x00080010,

    r1 = r2 = r3= 0x00000000

  - `LDMIA r0!, {r1-r3}`, or `LDMIA r0!, {r1, r2, r3}`
    - Register can be explicitly listed or use the "-" character

Pre-Condition for LDMIA Instruction

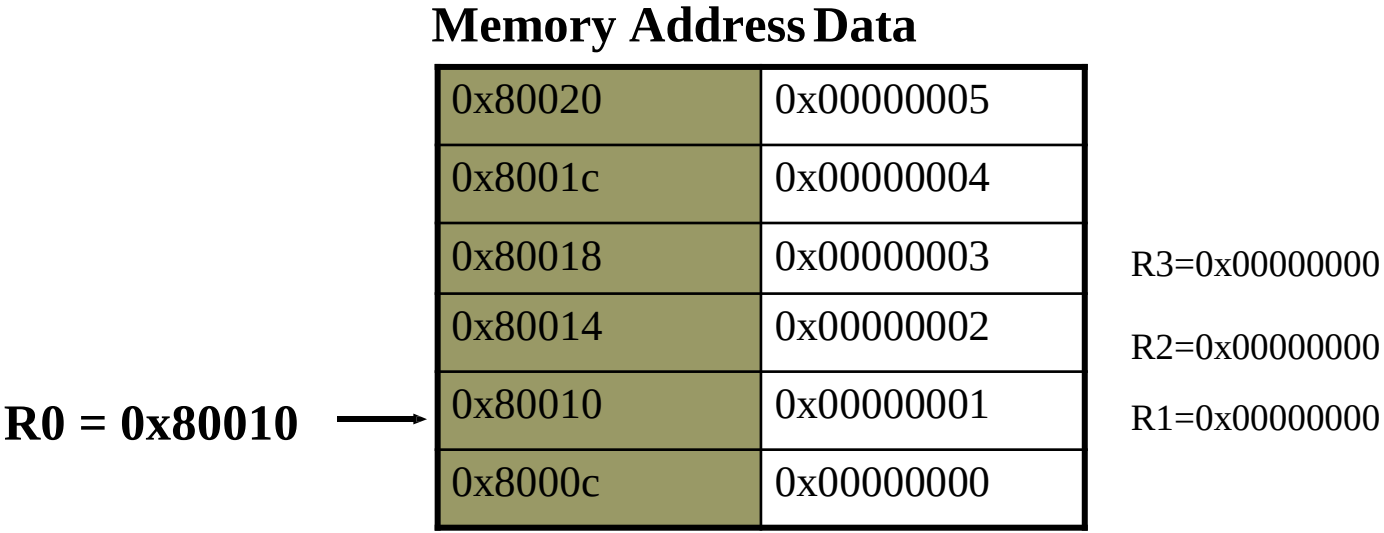


Figure 1

Post-Condition for LDMIA Instruction

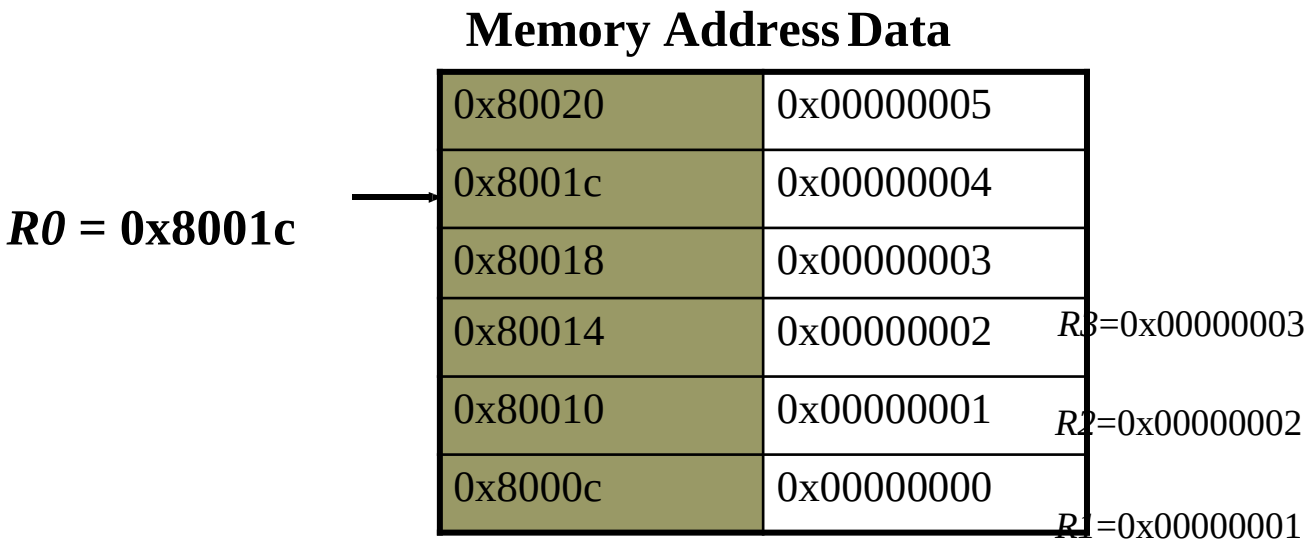


Figure 2